C-Source Reference for Operating the DS28E18 from Cortex-M4 GPIO Pins

MAXREFDES9002

Introduction

The DS28E18 is an easy-to-use bridge integrated circuit (IC) that operates as a 1-Wire® slave device to a host controller. The IC provides a 512-byte command sequencer loaded with commands and control data to operate an I^{2}C or SPI device interfaced to the DS28E18. Once loaded, the host controller sends a 1-Wire command to autonomously execute the sequence, power, and collect data from the attached peripherals. A subsequent 1-Wire command reads the collected data. The power for the attached sensors or peripherals is sourced from the 1-Wire line, enabling a very efficient solution to remotely power and control complex devices such as environmental sensors, analog-to-digital converters (ADCs), digital-to-analog converters (DACs), and display controllers. The MAXREFDES9002 is a reference design showcasing the DS28E18 that demonstrates how to implement a simple 1-Wire master using only a microcontroller's GPIO pins. To demonstrate this, this reference design provides a GPIO-based 1-Wire library designed for an ARM® Cortex® M4 microcontroller, such as the MAX32660, with example programs for interfacing an I^{2}C or SPI device connected to a DS28E18.

Features

- Power and communication to I^{2}C and SPI sensors through two wires
- 512-byte command sequencer for autonomous operation
- Up to 10mA at 3.3V sensor power derived from 1-Wire
- Up to 100m distance with the 1-Wire interface
- Example C-code demonstrates how to interface to I^{2}C and SPI sensors

Designed–Built–Tested

A simple, cost-efficient, I^{2}C and SPI bus extender using the DS28E18 is demonstrated for a simple sensor application designed for the MAX32660 host microcontroller. This reference design includes the following major components: a MAX32660, DS28E18, I^{2}C sensor, SPI sensor, and a couple of discrete components. This document describes the hardware shown in Figure 1 as well as its supplementing software. It provides a detailed, systematic technical guide to set up and understand the MAXREFDES9002 reference design. The system has been built and tested, details of which follow later in this document.

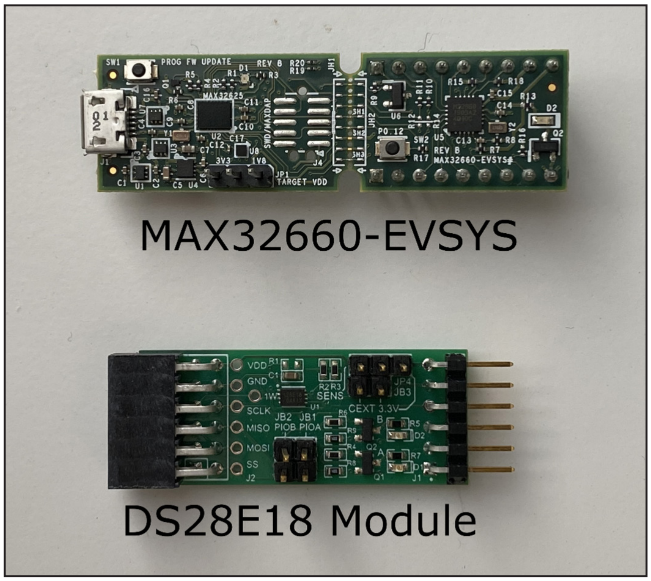


Figure 1. MAXREFDES9002 hardware.

[1] *1-Wire is a registered trademark of Maxim Integrated Products, Inc.*

Quick Start

This section includes a list of recommended equipment and instructions on how to set up the Windows -based PC for the C-Demo software.

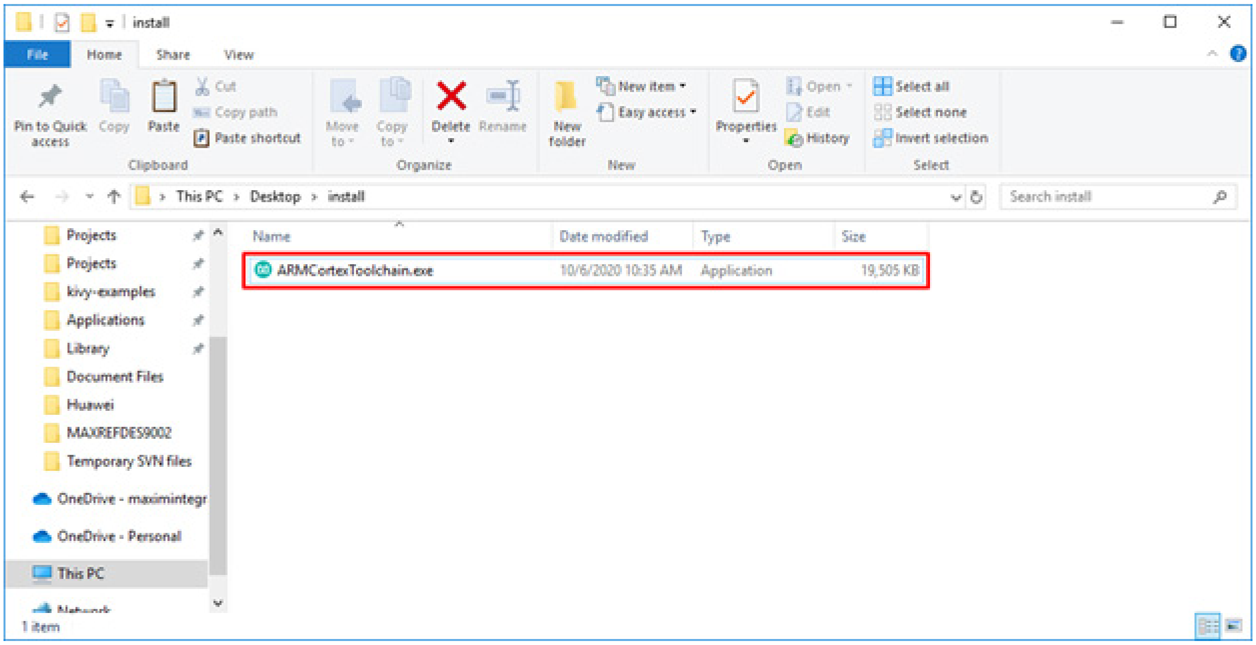
Required Equipment

- PC with a Windows-10, Windows-8, or Windows-7 operating system (64 bit or 32 bit), and a spare USB 2.0 or higher port
- Low-Power ARM[3] Micro Toolchain (Windows)
- C-Demo software

Procedure

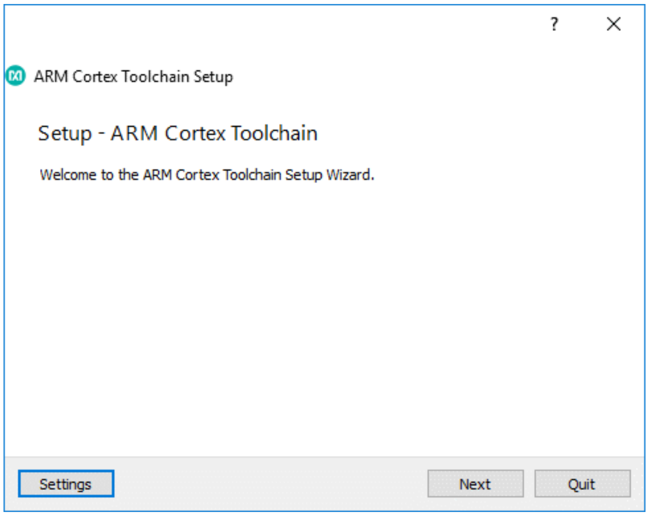
The reference design is fully assembled and tested. Follow these steps to set up the demo software:

1) Download the **ARMCortexToolchain.exe** file.
2) In a file viewer (Figure 2), double click **ARMCortex-Toolchain.exe** to begin the installation.
3) Follow the prompts on the setup wizard (Figure 3) to finish installation.



Name	Date modified	Type	Size
ARMCortexToolchain.exe	10/6/2020 10:35 AM	Application	19,505 KB

Figure 2. File viewer.



Setup - ARM Cortex Toolchain

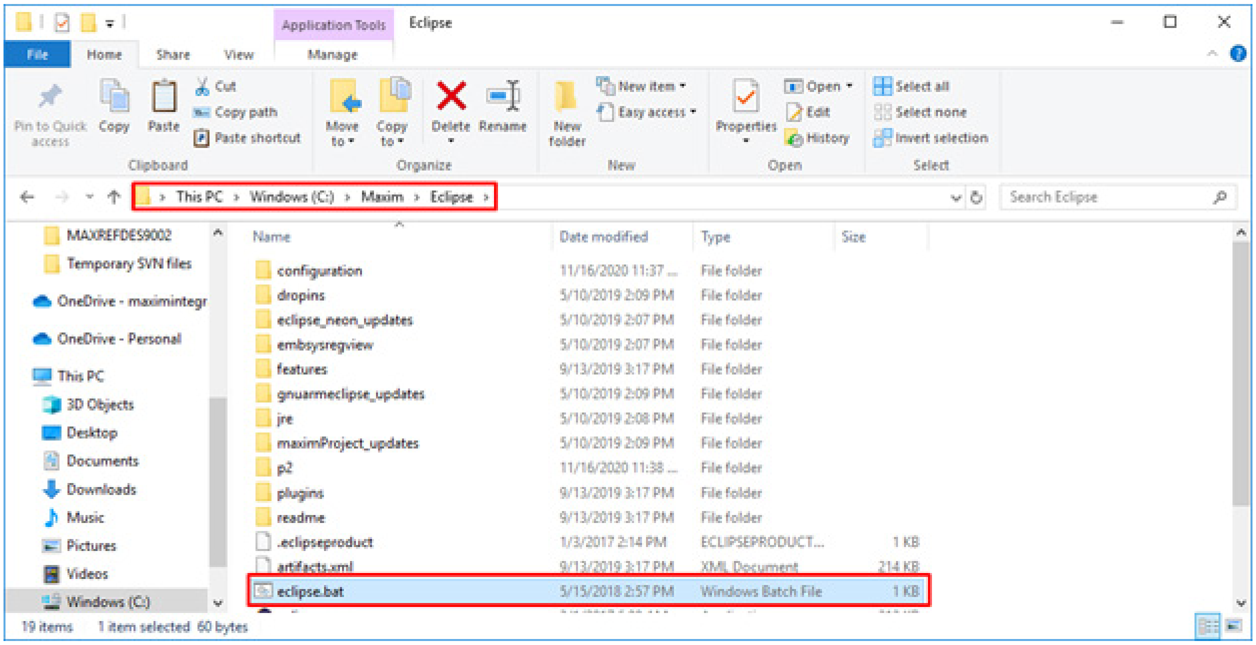
Welcome to the ARM Cortex Toolchain Setup Wizard.

Figure 3. Toolchain setup wizard.

[2] *Windows is a registered trademark and registered service mark of Microsoft Corporation.*
[3] *Arm and Cortex are registered trademarks of Arm Limited.*

4) Navigate to the toolchain's install directory, open the Eclipse folder and run Eclipse.bat to launch the Eclipse IDE (Figure 4).

5) Create a workspace in a desired location (Figure 5).

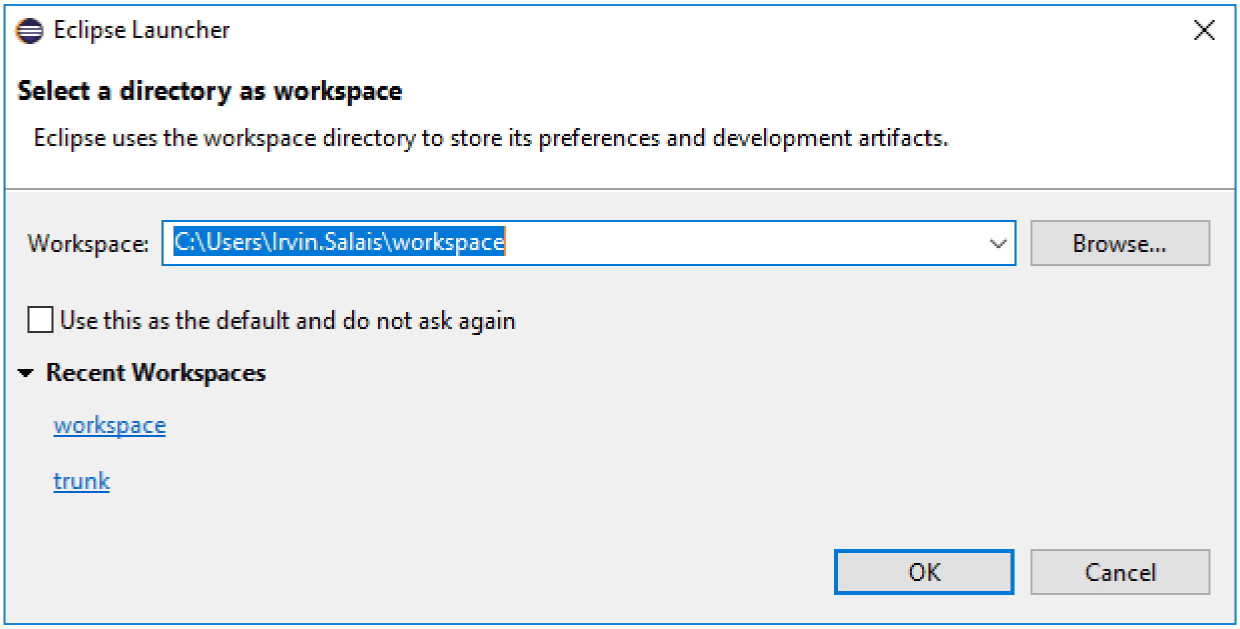


This PC › Windows (C:) › Maxim › Eclipse ›

Name	Date modified	Type	Size
configuration	11/16/2020 11:37 ...	File folder	
dropins	5/10/2019 2:09 PM	File folder	
eclipse_neon_updates	5/10/2019 2:07 PM	File folder	
embsysregview	5/10/2019 2:07 PM	File folder	
features	9/13/2019 3:17 PM	File folder	
gnuarmeclipse_updates	5/10/2019 2:09 PM	File folder	
jre	5/10/2019 2:08 PM	File folder	
maximProject_updates	5/10/2019 2:09 PM	File folder	
p2	11/16/2020 11:38 ...	File folder	
plugins	9/13/2019 3:17 PM	File folder	
readme	9/13/2019 3:17 PM	File folder	
.eclipseproduct	1/3/2017 2:14 PM	ECLIPSEPRODUCT...	1 KB
artifacts.xml	9/13/2019 3:17 PM	XML Document	214 KB
eclipse.bat	5/15/2018 2:57 PM	Windows Batch File	1 KB

19 items 1 item selected 60 bytes

Figure 4. Eclipse launch location.


Eclipse Launcher

Select a directory as workspace

Eclipse uses the workspace directory to store its preferences and development artifacts.

Workspace: C:\Users\Irvin.Salais\workspace

☐ Use this as the default and do not ask again

Recent Workspaces

workspace

trunk

Figure 5. Eclipse workspace creation.

6) Download and extract the **MAX32660_DS28E18_C_Demos-V1.0.0.zip** file in any location (Figure 6).

7) In Eclipse, go to **File\Import** and select **Existing Projects** into **Workspace** under the **General** folder. Click **Next >** and then **Browse** to open a directory browser. Navigate to the C-demo installation directory. Select the extracted folder containing the example programs and click **OK.** In the panel, check the checkbox next to the desired projects to import. Check **Copy projects into workspace** and click **Finish** (Figure 7).

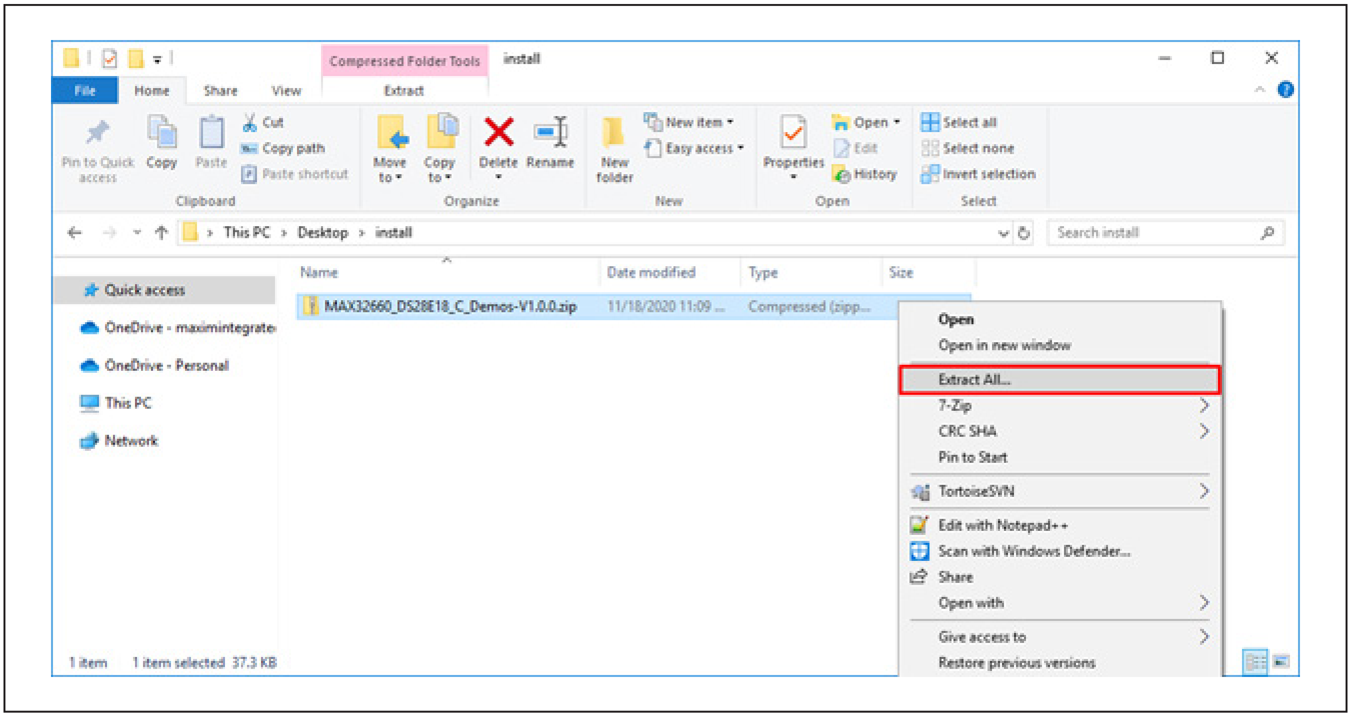
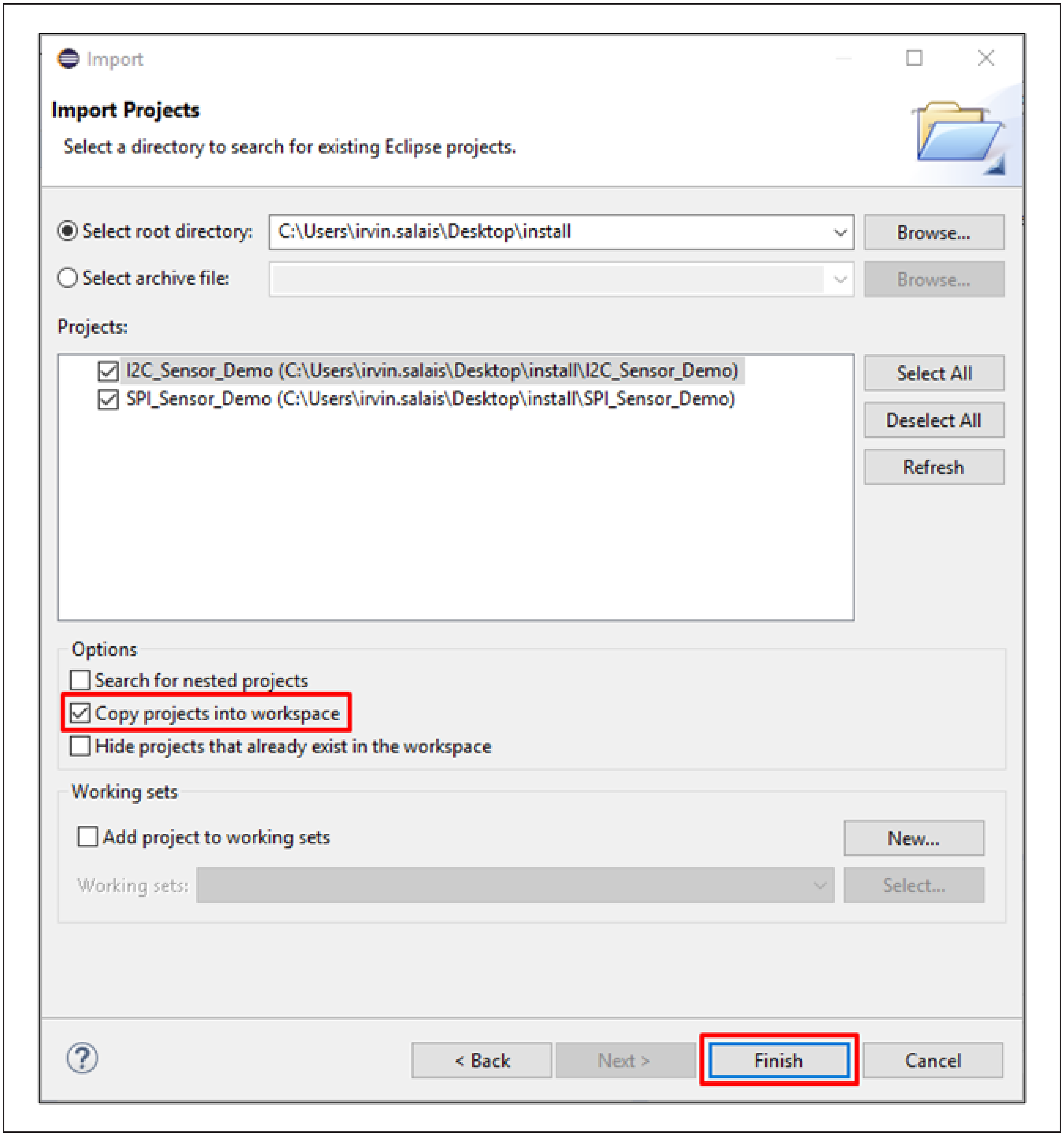


Figure 6. C-Demo extraction.



Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: C:\Users\irvin.salais\Desktop\install

○ Select archive file:

Projects:

☑ I2C_Sensor_Demo (C:\Users\irvin.salais\Desktop\install\I2C_Sensor_Demo)
☑ SPI_Sensor_Demo (C:\Users\irvin.salais\Desktop\install\SPI_Sensor_Demo)

Options

☐ Search for nested projects
☑ Copy projects into workspace
☐ Hide projects that already exist in the workspace

Working sets

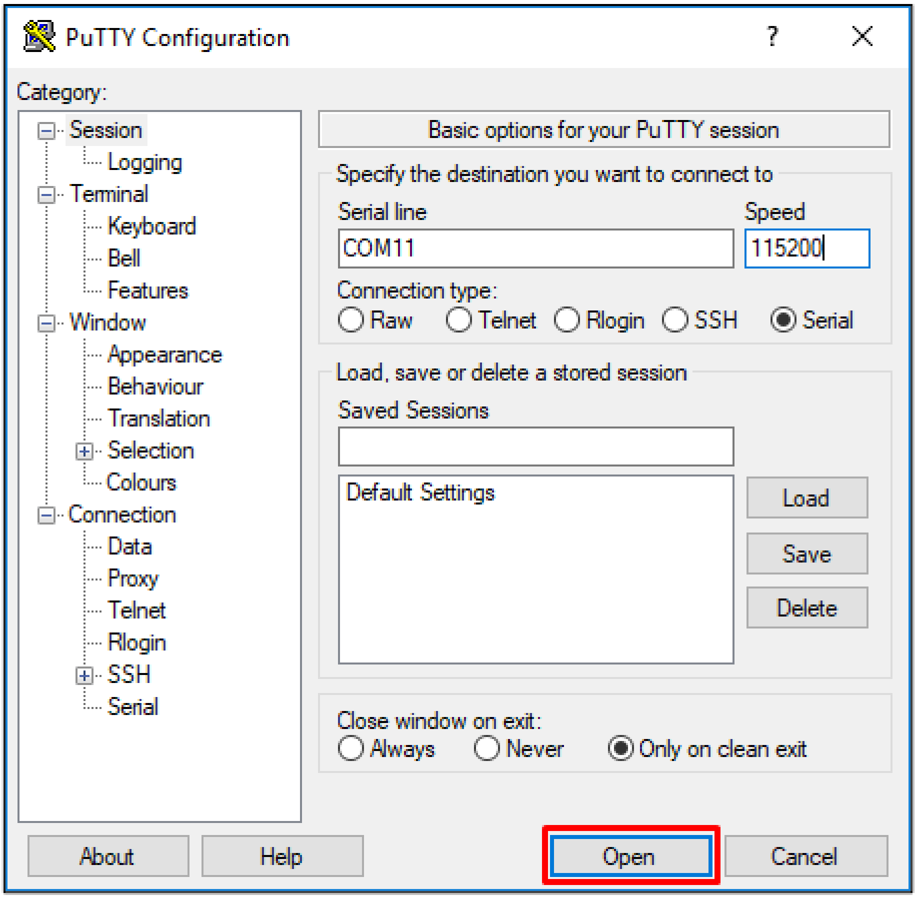
☐ Add project to working sets

Working sets:

Figure 7. Active clamp forward converter topology.

8) Open a serial console and connect it to the MAX32660's corresponding serial COM port (Figure 8).

9) Select the example program to run under the dropdown located next to the green Run button (Figure 9). The output is displayed on the serial console (Figure 10).



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: COM11

Speed: 115200

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

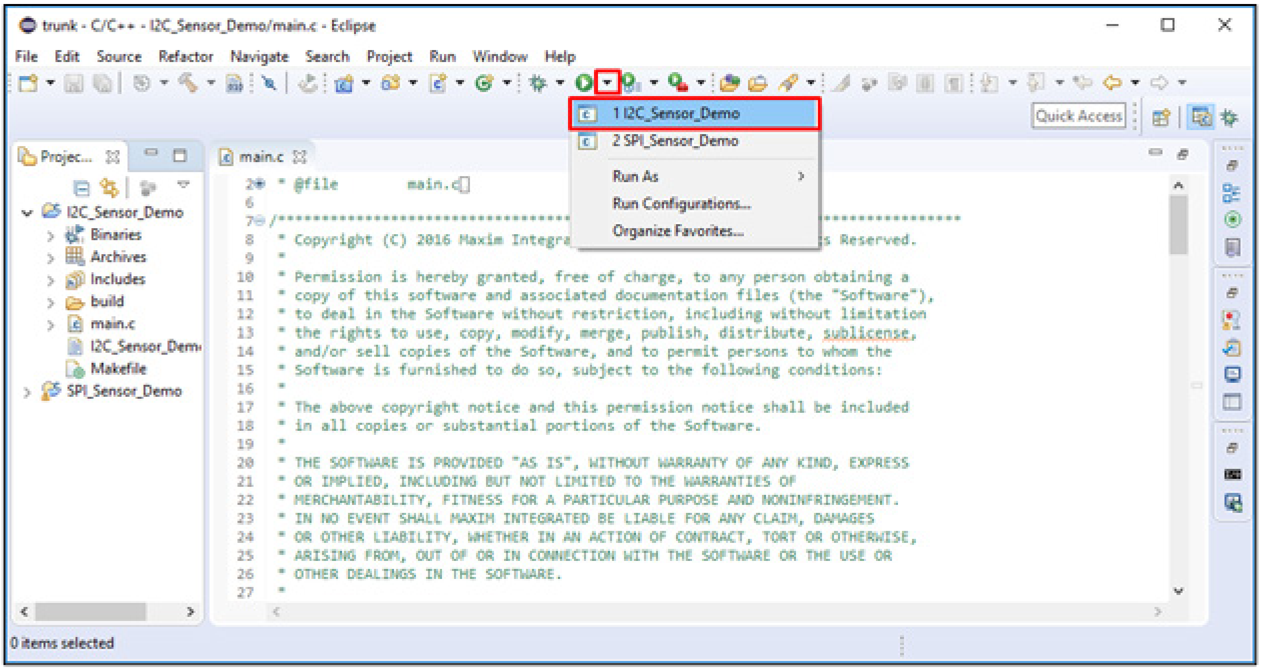
Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

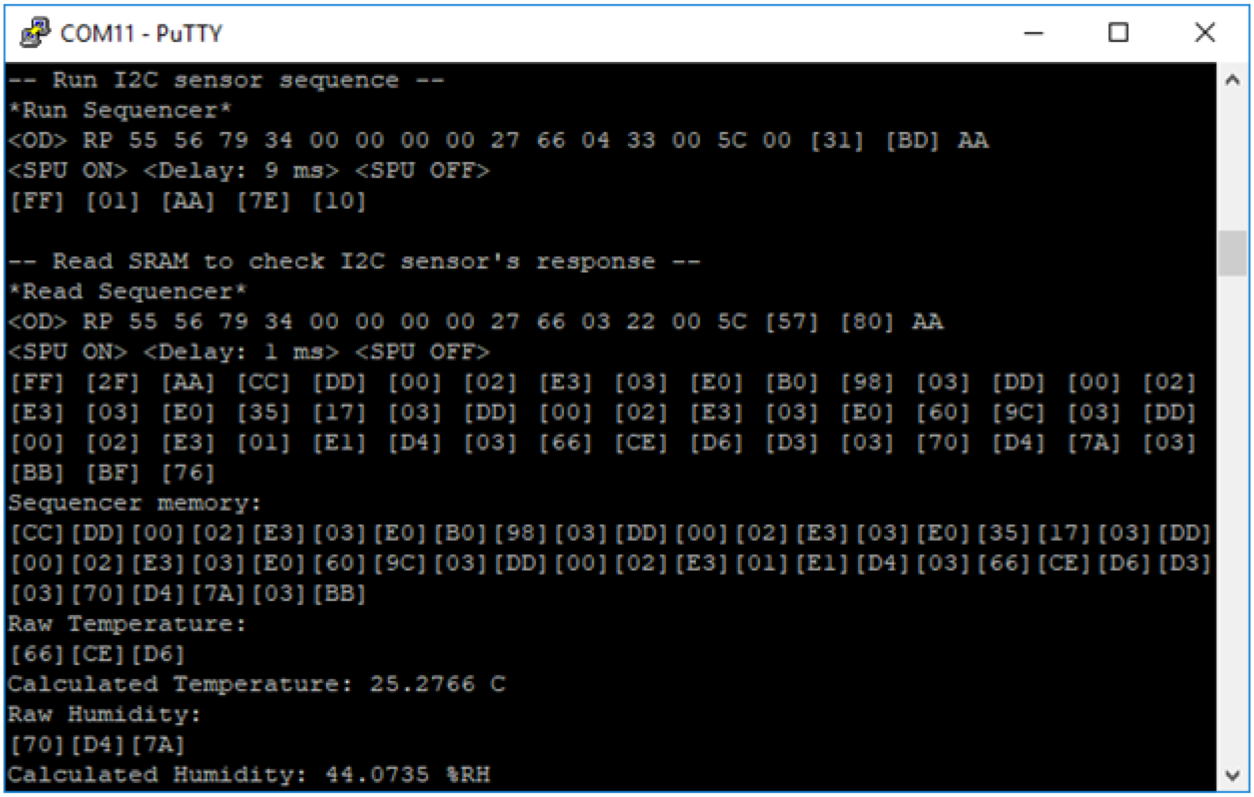
Figure 8. Opening a serial console.



```
 * @file       main.c

/*********************************                    ****************
 * Copyright (C) 2016 Maxim Integra                   s Reserved.
 *
 * Permission is hereby granted, free of charge, to any person obtaining a
 * copy of this software and associated documentation files (the "Software"),
 * to deal in the Software without restriction, including without limitation
 * the rights to use, copy, modify, merge, publish, distribute, sublicense,
 * and/or sell copies of the Software, and to permit persons to whom the
 * Software is furnished to do so, subject to the following conditions:
 *
 * The above copyright notice and this permission notice shall be included
 * in all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
 * OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
 * MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
 * IN NO EVENT SHALL MAXIM INTEGRATED BE LIABLE FOR ANY CLAIM, DAMAGES
 * OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
 * ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
 * OTHER DEALINGS IN THE SOFTWARE.
 *
```

Figure 9. Running I2C_Sensor_Demo example program.



```
-- Run I2C sensor sequence --
*Run Sequencer*
<OD> RP 55 56 79 34 00 00 00 00 27 66 04 33 00 5C 00 [31] [BD] AA
<SPU ON> <Delay: 9 ms> <SPU OFF>
[FF] [01] [AA] [7E] [10]

-- Read SRAM to check I2C sensor's response --
*Read Sequencer*
<OD> RP 55 56 79 34 00 00 00 00 27 66 03 22 00 5C [57] [80] AA
<SPU ON> <Delay: 1 ms> <SPU OFF>
[FF] [2F] [AA] [CC] [DD] [00] [02] [E3] [03] [E0] [B0] [98] [03] [DD] [00] [02]
[E3] [03] [E0] [35] [17] [03] [DD] [00] [02] [E3] [03] [E0] [60] [9C] [03] [DD]
[00] [02] [E3] [01] [E1] [D4] [03] [66] [CE] [D6] [D3] [03] [70] [D4] [7A] [03]
[BB] [BF] [76]
Sequencer memory:
[CC][DD][00][02][E3][03][E0][B0][98][03][DD][00][02][E3][03][E0][35][17][03][DD]
[00][02][E3][03][E0][60][9C][03][DD][00][02][E3][01][E1][D4][03][66][CE][D6][D3]
[03][70][D4][7A][03][BB]
Raw Temperature:
[66][CE][D6]
Calculated Temperature: 25.2766 C
Raw Humidity:
[70][D4][7A]
Calculated Humidity: 44.0735 %RH
```

Figure 10. Console output for I2C_Sensor_Demo example program.

Detailed Description of Hardware

Figures 11 and 12 show the main components and connections for the MAXREFDES9002 hardware. Connections from the DS28E18 to a compatible slave device varies depending on the device's interface. Figure 11 shows the typical hardware configurations for an I^2C slave device and Figure 12 for an SPI device. In either case, implementing the GPIO-driven 1-Wire master is the same. This reference design uses the MAX32660 microcontroller as the 1-Wire master. To drive the 1-Wire signal and strength, the MAX32660 uses the GPIO pins, P0_5, and P0_6, respectively. These pins are configured as open-drain outputs to create compatibility with the DS28E18 as the 1-Wire interface is of open-drain design. A low-impedance P-Channel MOSFET, Q1, is used to supply the 1-Wire bus with more current when demanded by the DS28E18, such as in an SPU event. The MAXREFDES9002 uses evaluation boards for the MAX32660 and DS28E18 to connect all the main components together. More details for each board are found in the *Design Resources* tab.

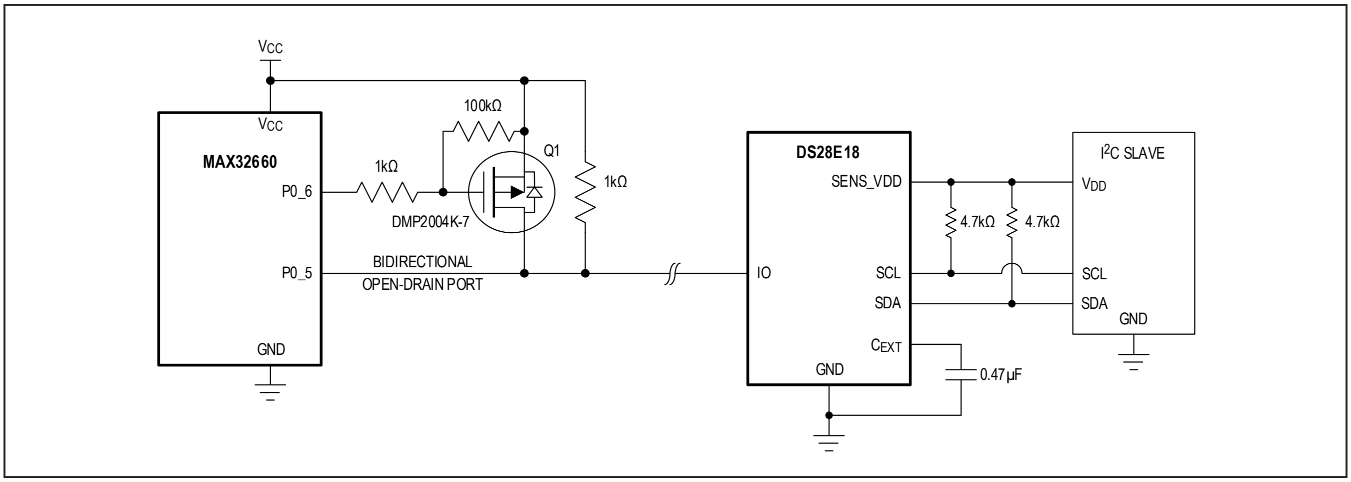


Figure 11. Typical MAXREFDES9002 hardware configurations for an attached I^2C sensor.

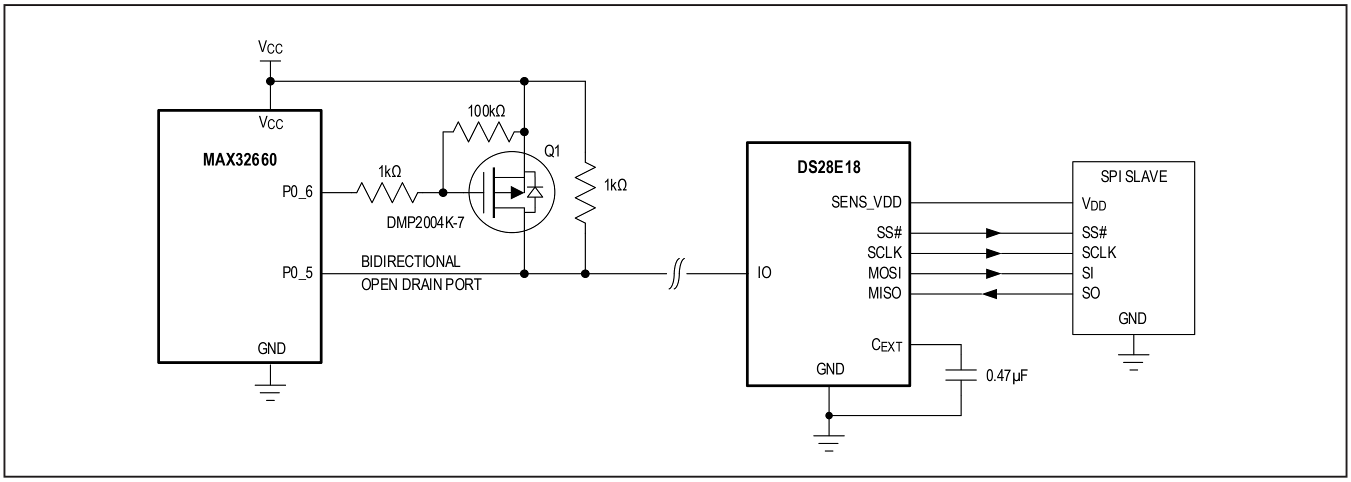


Figure 12. Typical MAXREFDES9002 hardware configurations for an attached SPI sensor.

Detailed Description of Software

The MAXREFDES9002 software consists of example code written in C used to interface with the hardware. This C-Demo software provides a 1-Wire API Developed from bit-banging the MAX32660's GPIO. See Table 1 for an overview of the 1-Wire API. This API allows the MAX32660 to interface with multiple 1-Wire slave devices as it includes all the necessary functions to control host 1-Wire communication for speeds in both Standard and Overdrive modes. Since it uses a GPIO to drive the 1-Wire line, the MAX32660 needs to operate its GPIO pins accordingly with respect to typical 1-Wire timings. The API does this by utilizing the MAX32660's peripheral timer, Timer0, to generate the precise timings. To set the line high, the microcontroller simply disables the output buffer of the pin corresponding to the 1-Wire bus (P0_5), allowing the external pullup resistor to drive the line high. Similarly, setting the line low can be achieved by simply enabling the output buffer in a low state. The C-Demo software also provides a DS28E18 API. This API makes it easy to exercise all the features of the DS28E18 and can be used to integrate multiple I^{2}C and SPI devices into a system. Two example programs are provided to demonstrate how to interface with any I^{2}C or SPI sensor connected to a DS28E18. The software is compatible with the Maxim toolchain, found in the *Design Resources* tab of the MAX32660 and can be directly imported into an Eclipse IDE workspace. See the *Quick Start*' section for details on how to set up the C-Demo software.

Design Resources

Download the complete set of **Design Resources** including schematics, bill of materials, PCB layout, and test files.

Table 1. 1-Wire API Overview

FUNCTION	DESCRIPTION
OneWire_Init	Sets up the MAX32660 as a 1-Wire master
OneWire_ResetPulse	Sends a 1-Wire Reset pulse down the 1-Wire bus
OneWire_WriteByte	Sends a specified byte down the 1-Wire bus
OneWire_WriteBytePower	Sends a specified byte down the 1-Wire bus and immediately enables the strong pullup (SPU)
OneWire_ReadByte	Requests a byte from the 1-Wire slave
OneWire_Search	Discovers multiple 1-Wire slaves found on the bus
OneWire_SetSpeed	Sets the 1-Wire Master speed between Standard and Overdrive

Table 2. DS28E18 API Overview

FUNCTION	DESCRIPTION
DS28E18_Init	Initializes all DS28E18 devices found on the 1-Wire bus
DS28E18_SetSpeed	Sets the DS28E18 speed between Standard and Overdrive
DS28E18_WriteSequencer	Writes specified data into the DS28E18 sequencer
DS28E18_ReadSequencer	Reads specified data from the DS28E18 sequencer
DS28E18_RunSequencer	Commands DS28E18 to execute specific I^{2}C/SPI sequences from its sequencer
DS28E18_WriteConfiguration	Configures the DS28E18's general configuration settings
DS28E18_ReadConfiguration	Reads the DS28E18's general configuration settings
DS28E18_WriteGpioConfiguration	Configures the DS28E18's GPIO configuration settings
DS28E18_ReadGpioConfiguration	Reads the DS28E18's GPIO configuration settings
DS28E18_DeviceStatus	Reads the DS28E18's Device Status
DS28E18_BuildPacket_X	Builds 1-Wire packet for the **X** Sequencer Command and stores it in a global array. The array can be used as a **DS28E18_Write_sequencer** input.

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	2/21	Initial release	—

Maxim Integrated
www.maximintegrated.com

Maxim Integrated cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim Integrated product. No circuit patent licenses are implied. Maxim Integrated reserves the right to change the circuitry and specifications without notice at any time. The parametric values (min and max limits) shown in the Electrical Characteristics table are guaranteed. Other parametric values quoted in this data sheet are provided for guidance.